CLAIMS

What is claimed is:

1. A method for implementing a programmable chip, the method comprising:

identifying first parameter information corresponding to a processor core, the first parameter information for configuring the processor core on the programmable chip;

identifying second parameter information corresponding to a peripheral, the second parameter information for configuring the peripheral on the programmable chip;

generating a logic description using the first and second parameter information, wherein the logic description provides logic information for implementing the processor core and the peripheral on the programmable chip.

2. The method of claim 1, wherein the peripheral is a peripheral component.

3. The method of claim 2, wherein the peripheral is a UART.

4. The method of claim 2, wherein the peripheral is a timer.

5. The method of claim 1, wherein the peripheral is a peripheral interface.

6. The method of claim 5, wherein the peripheral is a interface to off-chip memory.

7. The method of claim 1, wherein the logic description is a synthesizable logic file.

8. The method of claim 7, wherein the logic description is an HDL file.

9. The method of claim 1, wherein the logic description is a synthesized logic file.

10. The method of claim 9, wherein the logic description is an EDF file.

11. The method of claim 1, wherein the peripheral is a custom peripheral component.

12. The method of claim 1, wherein the peripheral is a custom peripheral interface.

13. The method of claim 1, wherein the logic information comprises device driver logic for implementing the peripheral on the programmable chip.

14. The method of claim 1, wherein generating the logic description using the first and second parameter information comprises generating connector logic for allowing interconnects between the processor core and the peripheral.

15. The method of claim 1, wherein generating connector logic comprises identifying the I/O ports associated with the processor core and the peripherals.

16. The method of claim 1, wherein identifying first and second parameter information comprises receiving first and second parameter information through a wizard.

17. The method of claim 1, wherein identifying first and second parameter information comprises receiving first and second parameter information through a subwizard, the subwizard spawned as a result of user interaction with the wizard.

18. A system for implementing a programmable chip, the system comprising:

memory;

a processor coupled with memory, the processor configured to identify first parameter information corresponding to a processor core, the first parameter information for configuring the processor core on the programmable chip, identify second parameter information corresponding to a peripheral, the second parameter information for configuring the peripheral on the programmable chip, and generate a logic description using the first and second parameter information, wherein the logic description provides logic information for implementing the processor core and the peripheral on the programmable chip.

19. The system of claim 18, wherein the peripheral is a peripheral component.

20. The system of claim 18, wherein the peripheral is a peripheral interface.

21. The system of claim 18 wherein the logic description is an HDL file.

22. The system of claim 18, wherein the logic description is a synthesized logic file.

23. The system of claim 18, wherein the peripheral is a custom peripheral

24. The system of claim 18, wherein the logic information comprises device driver logic for implementing the peripheral on the programmable chip.

25. A computer program product for implementing a programmable chip, the computer program product comprising:

computer code for identifying first parameter information corresponding to a processor core, the first parameter information for configuring the processor core on the programmable chip;

computer code for identifying second parameter information corresponding to a peripheral, the second parameter information for configuring the peripheral on the programmable chip; and

computer code for generating a logic description using the first and second parameter information, wherein the logic description provides logic information for implementing the processor core and the peripheral on the programmable chip.

26. The computer program product of claim 25, wherein the peripheral is a peripheral component.

27. The computer program product of claim 25, wherein the peripheral is a peripheral interface.

28. The computer program product of claim 25 wherein the logic description is an HDL file.

29. The computer program product of claim 25, wherein the logic description is a synthesized logic file.

30. The computer program product of claim 25, wherein the peripheral is a custom peripheral

31. The computer program product of claim 25, wherein the logic information comprises device driver logic for implementing the peripheral on the programmable chip.

32. A method for providing module information for download onto a programmable chip, the method comprising:

displaying a wizard to allow a user to enter first module information;

spawning a subwizard in response to a user action to allow the user to enter second module information;

providing first and second module information for download onto a programmable chip.

33. The method of claim 32, wherein the first module is a processor core.

34. The method of claim 32, wherein the first module is a peripheral.

35. The method of claim 32, wherein the programmable chip is a programmable logic device.

36. The method of claim 32, wherein the wizard and the subwizard are displayed in separate windows.

37. A system for providing module information for download onto a programmable chip, the system comprising:

memory;

a processor coupled with memory, the processor configured to display a wizard to allow a user to enter first module information, spawn a subwizard in response to a user action to allow the user to enter second module information, and provide first and second module information for download onto a programmable chip.

38. The system of claim 37, wherein the first module is a processor core.

39. The system of claim 37, wherein the first module is a peripheral.

40. The system of claim 37, wherein the programmable chip is a programmable logic device.

41. The system of claim 37, wherein the wizard and the subwizard are displayed in separate windows.

42. A method of implementing a programmable chip, the method comprising:

identifying first port information associated with a parameterized peripheral;

identifying second port information associated with a parameterized processor core;

generating a peripheral bus module to allow interconnections between the parameterized peripheral and the parameterized processor core, the interconnections allowing the implementation of the programmable chip.

43. The method of claim 42, wherein the peripheral is a peripheral component.

44. The method of claim 42, wherein the peripheral is a peripheral interface.

45. The method of claim 42, wherein the programmable chip is a programmable logic device.

46. A system of implementing a programmable chip, the system comprising:

memory;

a processor coupled with memory, the processor configured to identify first port information associated with a parameterized peripheral; identify second port information associated with a parameterized processor core; and generate a peripheral bus module to allow interconnections between the parameterized peripheral and the parameterized processor core, the interconnections allowing the implementation of the programmable chip.

47. The system of claim 46, wherein the peripheral is a peripheral component.

48. The system of claim 46, wherein the peripheral is a peripheral interface.

49. The system of claim 46, wherein the programmable chip is a programmable logic device.

50. A computer program product for implementing a programmable chip, the computer program product comprising:

computer code for identifying first port information associated with a parameterized peripheral;

computer code for identifying second port information associated with a parameterized processor core;

computer code for generating a peripheral bus module to allow interconnections between the parameterized peripheral and the parameterized processor core, the interconnections allowing the implementation of the programmable chip.

51. The computer program product of claim 50, wherein the peripheral is a peripheral component.

52. The computer program product of claim 50, wherein the peripheral is a peripheral interface.

53. The computer program product of claim 50, wherein the programmable chip is a programmable logic device.

54. A method for implementing a custom peripheral on a programmable logic device, the method comprising:

receiving first information associated with a custom peripheral;

receiving second information associated with a processor core;

generating a logic description using the first and second information to allow implementation of the custom peripheral on the programmable logic device.

55. The method of claim 54, wherein the custom peripheral is a custom peripheral component.

56. The method of claim 54, wherein the custom peripheral is a custom peripheral interface.

57. The method of claim 54, wherein the logic description is a synthesizable logic file.

58. The method of claim 57, wherein the logic description is an HDL file.

59. The method of claim 54, wherein the logic description is a synthesized logic file.

60. The method of claim 59, wherein the logic description is an EDF file.

61. A system for implementing a custom peripheral on a programmable logic device, the system comprising:

means for receiving first information associated with a custom peripheral;

means for receiving second information associated with a processor core;

means for generating a logic description using the first and second information to allow implementation of the custom peripheral on the programmable logic device.

62. The system of claim 61, wherein the custom peripheral is a custom peripheral component.

63. The system of claim 61, wherein the custom peripheral is a custom peripheral interface.

64. The system of claim 61, wherein the logic description is a synthesizable logic file.

65. The system of claim 64, wherein the logic description is an HDL file.

66. The system of claim 61, wherein the logic description is a synthesized logic file.

67. A computer program product for implementing a custom peripheral on a programmable logic device, the computer program product comprising:

computer code for receiving first information associated with a custom peripheral;

computer code for receiving second information associated with a processor core;

computer code for generating a logic description using the first and second information to allow implementation of the custom peripheral on the programmable logic device.

68. The computer program product of claim 67, wherein the custom peripheral is a custom peripheral component.

69. The computer program product of claim 67, wherein the custom peripheral is a custom peripheral interface.

70. The computer program product of claim 67, wherein the logic description is a synthesizable logic file.

71. The computer program product of claim 70, wherein the logic description is an HDL file.

72. The computer program product of claim 67, wherein the logic description is a synthesized logic file.

73. A method for implementing a device driver logic on a programmable logic device, the method comprising:

generating a logic description using a processor core module, a peripheral module, and a device driver associated with the peripheral module;

implementing the logic description on the programmable logic device, wherein implementing the logic description allows interaction between the peripheral module and the processor core module without any software device driver.

74. The method of claim 73, wherein the peripheral is a peripheral component.

75. The method of claim 73, wherein the peripheral is a peripheral interface.

76. The method of claim 73, wherein the logic description is an HDL file.

77. The method of claim 73, wherein the logic description is an EDF file.

78. A system for implementing a device driver logic on a programmable logic device, the system comprising:

means for generating a logic description using a processor core module, a peripheral module, and a device driver associated with the peripheral module;

means for implementing the logic description on the programmable logic device, wherein implementing the logic description allows interaction between the peripheral module and the processor core module without any software device driver.

79. The system of claim 78, wherein the peripheral is a peripheral component.

80. The system of claim 78, wherein the peripheral is a peripheral interface.

81. The system of claim 78, wherein the logic description is an HDL file.

82. The system of claim 78, wherein the logic description is an EDF file.

83. A computer program product for implementing a device driver logic on a programmable logic device, the computer program product comprising:

computer code for generating a logic description using a processor core module, a peripheral module, and a device driver associated with the peripheral module;

computer code for implementing the logic description on the programmable logic device, wherein implementing the logic description allows interaction between the peripheral module and the processor core module without any software device driver.

84. The system of claim 83, wherein the peripheral is a peripheral component.

85. The system of claim 83, wherein the peripheral is a peripheral interface.

86. The system of claim 83, wherein the logic description is an HDL file.

87. The system of claim 83, wherein the logic description is an EDF file.

88. A method for generating HDL from a general purpose programming language to implement a programmable chip, the method comprising:

identifying first parameter information corresponding to a processor core, the first parameter information for configuring the processor core on the programmable chip;

identifying second parameter information corresponding to a peripheral, the second parameter information for configuring the peripheral on the programmable chip;

generating HDL from program code associated with the processor core and peripheral using the first and second parameter information, wherein the HDL allows implementation of the processor core and the peripheral on the programmable chip.

89. The method of claim 88, wherein the peripheral is a peripheral component.

90. The method of claim 88, wherein the peripheral is a peripheral interface.

91. A system for generating HDL from a general purpose programming language to implement a programmable chip, the system comprising:

memory;

a processor coupled with memory, the processor configured to identify first parameter information corresponding to a processor core, the first parameter information for configuring the processor core on the programmable chip, identify second parameter information corresponding to a peripheral, the second parameter information for configuring the peripheral on the programmable chip, and generate HDL from program code associated with the processor core and peripheral using the first and second parameter information, wherein the HDL allows implementation of the processor core and the peripheral on the programmable chip.

92. The system of claim 91, wherein the peripheral is a peripheral component.

93. The system of claim 91, wherein the peripheral is a peripheral interface.